A

El manual del usuario del DSP56303EVM

(En este Apéndice se incluyen algunos de los aspectos más relevantes del manual del usuario del DSP56303EVM. Éste se ha incluido por el hecho de que el DSP56002 ha sido declarado oficialmente descontinuado a partir de agosto del 2002. Aunque seguramente se pueden encontrar todavía algunos sistemas de evaluación basados en el procesador de señales DSP56002, se considera oportuno actualizar las opciones de trabajo en el laboratorio incluyendo una nueva versión de la familia DSP56K de procesadores de señales.

El material está basado en el *DSP56303EVM User's Manual*, con permiso de Freescale Semiconductors, Inc. El documento se puede ordenar como DSP56303EVMUM/D Rev. 3.4, 12/1999.)

A.1. Guía de Inicio Rápido

En esta sección se resume el contenido del módulo de evaluación y los requerimientos adicionales, también se da información para una rápida instalación y cómo efectuar una prueba del sistema. En las siguientes secciones el manual da detalles sobre el diseño y operación del DSP56303.

A.1.1. Equipo

En la siguiente sección se da una lista del equipo requerido para el uso del módulo de evaluación del DSP56303 (DSP56303EVM), parte de este equipo viene junto con el módulo, y otra parte debe suministrarla el usuario.

Lo que Ud. obtiene con el DSP56303EVM

El siguiente material viene con el DSP56303EVM:

- Tarjeta del módulo de evaluación DSP56303.
- Folleto de producto DSP56303EVM.
- Manual del usuario del DSP56303EVM.
- Especificaciones de producto DSP56303, revisión 1.03.
- Hoja de errata del circuito integrado (CI, o *Chip*) DSP56303.
- Hoja de datos del Codec de audio multimedia *Crystal Semiconductor* CS4218 de 16-bits.
- Documentación técnica en CD-ROM incluyendo los siguientes documentos:
  - Manual de la familia DSP56300.
  - Manual del usuario DSP56303.
  - Hoja de datos técnicos del DSP56303.
- El *software* (programas de aplicación) requerido:
  - depurador gráfico (GUI Debugger) de Domain Technologies (1 CD)
  - Programa de cómputo Ensamblador/encadenador de Freescale Semiconductors, Inc. (1 CD)

Lo que Ud. necesita suministrar

El usuario debe proveer lo siguiente:

- Computadora personal (PC) (pentium-90 MHz o mayor) con
  - Windows95 o NT,
  - 16 Mbytes de memoria como mínimo en Windows95,
  - 32 Mbytes de memoria como mínimo en Windows NT,
  - disco flexible de 31/2 pulgadas,
  - lector de CD,
  - disco duro con 20 Mbytes de espacio libre,
  - ratón,
  - puerto serie RS-232 que soporte una tasa de transferencia de datos 9,600-115,200 bits-por-segundo.

- Cable para interfase RS-232 (DB9 macho a DB9 hembra).
- Fuente de Poder, 7-9 V AC o DC con un conector de fuente de 2.1 mm.
- Fuente de señales de audio (reproductor de cintas, radio, reproductor de CDs, etc.).
- Cable de interfase para audio con conectores macho estero de 1/8 de pulgada (de diámetro).
- Audífonos.

A.1.2. Procedimiento de instalación

La instalación requiere de los siguientes cuatro pasos básicos:

1. Preparar la tarjeta DSP56303EVM.
2. Conectar la tarjeta a la PC y a la fuente de alimentación.
3. Instalación de las aplicaciones (software).
4. Prueba de la instalación.

Precaución

Debido a que todos los componentes electrónicos
son sensibles a los efectos de descarga
electrostática (DE), deben observarse procedimientos
adecuados cuando se manipulan todos los
componentes de este paquete y dentro de la PC
Para minimizar la posibilidad de daño debido a DE
sigua el siguiente procedimiento:

a) Simpre maneje todos los componentes sensibles a ED
sólo desde su área protejida, de preferencia
en el lab con piso antiestático y mesas de trabajo.

b) Use cintas aterrizadas adheridas a su muñeca.

c) No retire los componentes de su envoltura antiestática
hasta que se requiera para su instalación.

d) Siempre transporte los componentes sensibles dentro
de su envoltura antiestática.

Preparación del DSP56303EVM

Localice los bloques de puentes J1, J4, J5 y J8, como se muestra en la figura A.1. Para el bloque J1, asegúrese de que tenga los puentes conectando las terminales 3-4, y 5-6. Para los bloques J4, J5, y J8 asegúrese que todas las posiciones en cada bloque estén conectadas con puentes. Estos puentes realizan las siguientes funciones:

- J1 controla el modo de operación del DSP56303.
- J4 y J5 controlan la interfase entre el codec de audio y la interfase serie síncrona mejorada (ESSIO) del DSP56303.
- J8 controla la interfase entre el puerto JTAG/OnCE del DSP56303 y la interfase serie síncrona (SSI) del DSP56002.

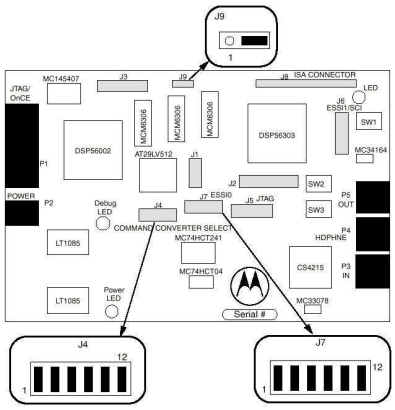

Figura A.1: Ubicación de componentes del DSP56303EVM.

Conexión del DSP56303EVM a la PC y a la fuente de poder

La figura A.2 muestra el diagrama de interconexión para conectar la PC y la fuente de poder externa a la tarjeta DSP56303EVM.

Emplee los siguientes pasos para completar la conexión de cables:

1. Conecte extremo del cable RS-232 con el conector DB9 macho al puerto RS-232 de la PC.
2. Conecte el extremo del cable DB9 al conector P2 del DSP56303 que se muestra en la figura A.1. Esto provee la conexión que permite a la PC controlar el funcionamiento de la tarjeta.
3. Asegúrese de que la fuente de poder 7-9 V no esté alimentada con energía.

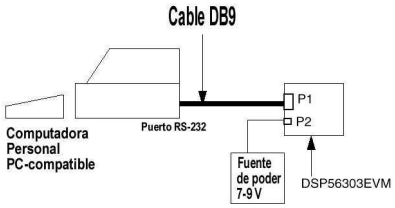

Figura A.2: Conexión de cables al DSP56303EVM.

4. Conecte la salida de la fuente de poder, el conector de 2.1, a P3 de la tarjeta DSP56303, mostrado en la figura A.1.

5. Aplique energía a la fuente de poder. El diodo emisor de luz verde (LED) se enciende cuando la energía esté aplicada correctamente.

A.2. Ejemplo con programa de prueba

Esta sección contiene un ejemplo que ilustra cómo desarrollar un programa muy sencillo para el DSP56303. Este ejemplo es para los usuarios con muy poca o nula experiencia con las herramientas de desarrollo de DSPs. El ejemplo demuestra la forma de los programas en ensamblador, da instrucciones de cómo ensamblar programas, y muestra cómo el depurador puede verificar la operación de programas.

La figura A.3 muestra en un diagrama de flujo el proceso de desarrollo de programas en ensamblador. Los bloques redondeados representan los archivos en código ensamblador y en código objeto. Los bloques en blanco representan programa de computadora (*software*) para el ensamblado de programas. Los bloques en gris representan los productos de circuitería (*hardware*).

Las siguientes secciones muestran información básica sobre el programa en código ensamblador, el programa ensamblador, el programa encadenador (*linker*) y los archivos objeto. Para información detallada sobre estos temas, consulte los manuales del ensamblador y del encadenador incluidos en el paquete de programas DSP CLAS, el cual está disponible a través de las oficinas o distribuidores de Freescale Semiconductors, Inc. La documentación también está disponible a través de `Motorola Wireless internet URL`: `http://www.mot.com/SPS/DSP/documentation`.

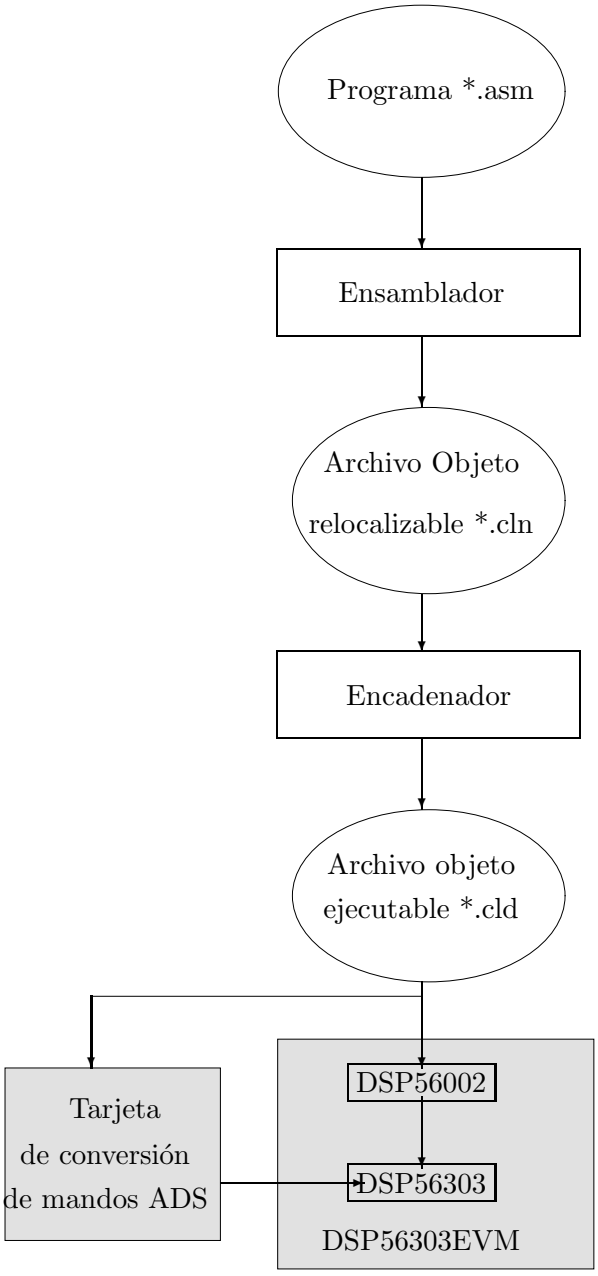

Figura A.3: Flujo del proceso de desarrollo.

A.2.1. Escritura del programa

Las siguientes secciones describen el formato de los enunciados del lenguaje ensamblador y dan un ejemplo de un programa en código ensamblador.

Formato de enunciados fuente

Los programas escritos en lenguaje ensamblador consisten de una secuencia de enunciados fuente. Cada enunciado fuente puede incluir hasta seis campos separados por uno o más espacios o TAB (tabulador): un campo de etiquetas, un campo de operación, un campo de operandos, hasta dos campos de transferencia de datos, y un campo de comentarios. Por ejemplo, el enunciado incluido en la tabla A.1, muestra los seis campos posibles:

Tabla A.1: Ejemplo de enunciado fuente.

trm	mac	x0,y0,a	x:(r0)+,x0	y:(r4)+,y0	;Texto
↓	↓	↓	↓	↓	↓
Etiqueta	Operación	Operando	T. datos X	T.datos Y	Comentario

El campo etiqueta

El campo etiqueta es el primer campo de un enunciado fuente y puede tener una de las siguientes formas:

- Un espacio o tabulador como primer carácter sobre una línea generalmente indica que el campo de etiqueta es vacío y que la línea no tiene etiqueta.
- Un carácter alfanumérico como primer carácter indica que la línea contiene un símbolo llamado etiqueta.
- Un guión bajo como primer carácter indica que la etiqueta es local.

Con la excepción de algunas directivas, a una etiqueta se le asigna el valor del contador de localidades de la primera palabra de la instrucción o dato que está siendo ensamblado. Una línea que consiste sólo de una etiqueta es una línea válida y se le asigna el valor del contador de localidades a la etiqueta.

El campo de operación

El campo de operación aparece después del campo de etiqueta y debe estar precedido de, al menos, un espacio o tabulador. El ingreso en el campo de operación puede ser uno de estos tres tipos:

- **Opcode**- mnemónicos que corresponden directamente a instrucciones de machina del DSP.
- **Directiva**- códigos de operación especial conocidos por el ensamblador que controlan el proceso de ensamblado.
- **Macro call**- Llamada a un macro definido previamente, el cual es insertado en el lugar del llamado al macro.

Campo de operando

La interpretación del campo de operando depende del contenido del campo de operación. El campo de operando, si está presente, debe seguir al campo de operación y debe estar precedido de por lo menos un espacio o tabulador (tab).

Campos de transferencia de datos

La mayoría de los códigos de operación especifican una o más transferencias de datos que ocurrirán durante la ejecución de la instrucción. Estas transferencias de datos son indicadas por dos operandos de modo de direccionamiento separados por una coma, sin insertar espacios en blanco. Si se especifican dos transferencias de datos, estos deben estar separados por uno o más espacios en blanco o tabs. Para una completa discusión de los modos de direccionamiento aplicables a las especificaciones de transferencia véase el *DSP56300 Family Manual*.

El campo de comentarios

Los comentarios no tienen significado para el ensamblador, pero pueden incluirse en el archivo fuente con propósitos de documentación. Un campo de comentarios está compuesto de cualesquiera caracteres que sean precedidos por un punto y coma.

A.2.2. Ejemplo de programa

El programa de ejemplo discutido en esta sección toma dos listas de datos, una en la memoria X y la otra en la memoria Y, y calcula la suma de los productos de las dos listas. El cálculo de la suma de productos es la base para muchas funciones de PDS. Por lo tanto, el DSP56303 tiene una instrucción especial, multiplica-acumula (MAC), que multiplica dos valores y suma el resultado al contenido de un acumulador.

```
          Ejemplo de un código DSP56303 sencillo
-----------------------------------------------------
;*****************************************************
; UN PROGRAMA SENCILLO: CÁLCULO DE LA SUMA DE PRODUCTOS
;*****************************************************
```

```
PBASE   EQU $100    ;Instruye al ensamblador reemplazar cada
                    ;ocurrencia de PBASE  por $100
XBASE   EQU $0      ;Usado para definir la posición de los
                    ;datos en la memoria X
YBASE   EQU $0      ;Usado para definir la posición de los
                    ;datos en la memoria Y
;*********************************************************
;MEMORIA X
;*********************************************************
        org x:XBASE ;indica al ensamblador que nos referimos
                    ;a la memoria X comenzando
                    ;en la localidad XBASE
list1   dc  $475638,$738301,$92673a,$898978,$091271,$f25067
        dc  $987153,$3a8761,$987237,$34b852,$734623,$233763
        dc  $f76756,$423423,$324732,$f40029
;*********************************************************
;MEMORIA Y
;*********************************************************
        org Y:YBASE ;indica al ensamblador que nos referimos
                    ;a la memoria Y iniciando en
                    ;la localidad YBASE
list2   dc  $f98734,$800000,$fedcba,$487327,$957572,$369856
        dc  $247978,$8a3407,$734546,$344787,$938482,$304f82
        dc  $123456,$657784,$567123,$675634
;*********************************************************
;PROGRAMA
;*********************************************************
        org p:0     ;pone el siguiente programa en la memoria
                    ;de programa comenzando en la localidad 0
        jmp begin   ;p:0 es el vector de re-inicio i.e. en
                    ;donde el DSP busca instrucciones después
                    ;del re-inicio (reset)
        org p:PBASE ;Comienza programa principal en p:PBASE
begin
        move #list1,r0 ;prepara apuntador al inicio de list1
        move #list2,r4 ;prepara apuntador al inicio de list2
        clr  a         ;borra el contenido del acumulador a
        move x:(r0)+,x0  y:(r4)+,y0
                       ;carga en x0 el valor de la memoria X
                       ;apuntada por el contenido de r0 y
                       ;post-incrementa r0
                       ;carga en y0 el valor de la memoria Y
                       ;apuntada por el contenido de r4 y
                       ;post-incrementa r4
        do   #15,endloop ; ejecuta 15 veces
        mac  x0,y0,a  x:(r0)+,x0  y:(r4)+,y0
                       ;multiplica y acumula, y carga los
                       ;siguientes valores
endloop jmp  *         ;esto es equivalente a
                       ;etiqueta jmp etiqueta
```

```
                        ; y es, por lo tanto, un lazo vacío
                        ;infinito
;*********************************************************
;FIN DEL PROGRAMA SENCILLO
;*********************************************************
```

Edite y grabe un archivo con el contenido del ejemplo y asigne el nombre `ejemplo.asm`.

A.3. Ensamblado del programa

Las siguientes secciones describen el formato de los mandos del ensamblador, la lista de caracteres especiales y directivas del ensamblador, y dan instrucciones par ensamblar el programa de ejemplo.

A.3.1. Formato del mando del ensamblador

El ensamblador DSP viene incluido con el DSP56303EVM en el CD de herramientas y se puede instalar siguiendo las instrucciones que vienen en el archivo `Read me` en el CD. El ensamblador DSP es un programa que traduce los enunciados fuente en lenguaje ensamblador en programas objeto compatibles con el DSP56303. El formato general en la línea de mandos para invocar al ensamblador es `asm56300 [opciones] <nombre-de-archivos>` donde `asm56300` es el nombre del programa ensamblador del DSP, y `<nombre-de-archivos>` es una lista de programas en lenguaje ensamblador a ser ensamblados.

A.3.2. Opciones del ensamblador

La tabla A.2 describe las opciones del ensamblador. Para evitar ambigüedad, los argumentos de las opciones deben seguir inmediatamente la letra de la opción sin blancos entre ellos.

A.3.3. Directivas del ensamblador

Además del juego de instrucciones, los programas en ensamblador pueden contener mnemónicos que son directivas para especificar acciones auxiliares a ser realizadas por el ensamblador. Éstas son las directivas del ensamblador, y no siempre son traducidas a lenguaje de máquina. Las siguientes secciones describen brevemente varios tipos de directivas del lenguaje ensamblador.

Caracteres significativos del ensamblador

A continuación se muestran las secuencias de uno y dos caracteres que son significativas para el ensamblador:

```
;       delimitador de comentarios
;;      delimitador de comentario sin reportar
\       carácter continuación de línea o
        argumento falso (dummy) de macro de
        concatenación de operadores
?       operador macro de substitución de valor
%       operador macro de substitución de valor hexadecimal
^       operador macro de reescritura de etiqueta local
''      cadena macro delimitador o cadena de expansión
        de carácter DEFINE
@       delimitador de función
*       substitución local de contador
++      operador de concatenación de cadena
[]      delimitador de subcadena
<<      operador para forzar el modo de
        direccionamiento corto de E/S
<       operador para forzar el modo de
        direccionamiento corto
>       operador para forzar el modo de
        direccionamiento largo
#       operador de modo de direccionamiento inmediato
#<      operador para forzar el modo de
        direccionamiento inmediato corto
#>      operador para forzar el modo de
        direccionamiento inmediato largo
```

Tabla A.2: Opciones del ensamblador

Opción	Descripción
-A	Pone al ensamblador en modo absoluto y genera un archivo objeto absoluto cuando se da la opción **-B** en la línea de mandos. Por omisión, el ensamblador produce un archivo objeto relocalizable que es subsecuentemente procesado por el Encadenador (*linker*) DSP de Motorola.
-B`<archobj>`	Especifica que sea creado un archivo objeto para la salida del ensamblador. `<archobj>` puede ser cualquier nombre legal según el sistema operativo, incluyendo una trayectoria la cual es opcional. El tipo de archivo objeto depende del modo de operación del ensamblador. Si se da la opción **-A** en la línea de mandos, el ensamblador opera en modo absoluto y genera un archivo objeto absoluto (`.cld`). Si no hay una opción **-A**, el ensamblador opera en modo relativo y crea un archivo objeto relocalizable (`.cln`). Si no se especifica la opción **-B**, el ensamblador no genera un archivo objeto. Si no se especifica `<archobj>`, el ensamblador usa el nombre base (nombre de archivo sin extensión) del primer archivo encontrado en la lista de archivos fuente de entrada y agrega el tipo de archivo apropiado (`.cln` o `.cld`) al nombre base. La opción **-B** debe especificarse sólo una vez. Ejemplo: `asm56300 -Bfiltro main.asm fft.asm fio.asm` Este ejemplo ensambla los archivos `main.asm`, `fft.asm`, y `fio.asm` juntos para producir el archivo objeto relocalizable `filtro.cln`
-D`<simbolo>` `<cadena>`	Reemplaza todas las ocurrencias de `<simbolo>` con `<cadena>` en los archivos fuente que sean ensamblados Ejemplo: `asm56300 -DPOINTS 16 prog.asm` Reemplaza todas las ocurrencias del símbolo POINTS en el programa `prog.asm` por la cadena '16'.

Tabla A.3: Opciones del ensamblador (cont.)

Opción	Descripción
-EA<errfil> o -EW<errfil>	Permite reasignar el archivo estándar de salida de errores en máquinas que no soportan redireccionar desde la línea de mandos. <errfil> debe estar presente como un argumento pero puede ser cualquier nombre de archivo válido en el sistema operativo, incluyendo una trayectoria válida. La opción **-EA** causa que la cadena estándar de error sea escrita en <errfil>;. Si <errfil> existe, la cadena de salida se agrega al final del archivo. La opción **-EW** también escribe la cadena estándar de error a <errfil>; si <errfil> existe, es reescrito. Ejemplo: `asm56300 -EWerrors prog.asm` Redirige la salida estándar al archivo `errors`. Si el archivo existe, es reescrito.
-F<argfil>	Indica que el ensamblador debe leer la entrada de la línea de mandos desde <argfil>, el cual puede ser cualquier nombre de archivo legal en el sistema operativo, incluyendo un trayectoria opcional. <argfil> es un archivo de texto que contiene más opciones, argumentos, y nombres de archivos que son pasados al ensamblador. Los argumentos en el archivo necesitan estar separados solamente por un espacio. Un punto y coma (;) sobre una línea que sigue a un espacio en blanco hace que el resto de la línea sea un comentario. Ejemplo: '+asm56300 -Fopts.cmd+ Invoca al ensamblador y toma las opciones de la línea de mandos y nombre de los archivos fuente desde el archivo de mandos `opts.cmd`
-G	Envía información del número de línea del archivo fuente al archivo objeto. Esta opción es válida sólo conjuntamente con la opción de la línea de mandos **-B**. Los programas depuradores pueden usar la información del número de línea generado para proveer depuración a nivel de código fuente. Ejemplo: `asm56300 -B -Gmiprog.asm` Ensambla el archivo `miprog.asm` y envía la información del número de línea del archivo fuente al archivo objeto resultante `miprog.cln`

Tabla A.4: Opciones del ensamblador (cont.)

Opción	Descripción
-I`<trayectoria>`	Causa que el ensamblador busque, en el directorio definido por `<trayectoria>`, cualquier archivo no encontrado en el directorio corriente o actual. `<trayectoria>` puede ser cualquier nombre de trayectoria (directorio) legal del sistema operativo. Ejemplo: `asm56300 -I\verb+` Usa la convención de la PC IBM en la trayectoria y causa que el ensamblador prefije cualquier archivo incluido (con *include*) no encontrado en el directorio actual con la trayectoria `\proyecto\trayectoria`.
-L`<lstfil>`	Especifica crear un archivo de listado para la salida de ensamblador. `<lstfil>` puede ser cualquier nombre de archivo legal en el sistema operativo, incluyendo opcionalmente el nombre de la trayectoria. Si no se especifica `<lstfil>`, el ensamblador usa el nombre base (nombre del archivo sin la extensión) del primer archivo encontrado en la lista de archivos fuente de entrada y agrega la extensión `.lst` al nombre base. La opción **-L** se especifica sólo una vez. Ejemplo: `asm 56300 -Lfiltro.asm gauss.asm` Ensambla los archivos `filtro.asm` y `gauss.asm` juntos y produce un archivo de listado. Debido a que no se dio el nombre del archivo, el archivo de salida es nombrado usando el nombre base del primer archivo fuente encontrado, en este caso filtro, y el archivo de listado es llamado `filtro.lst`.
-M`<trayectoria>`	Causa que el ensamblador busque en el directorio definido por `<trayectoria>` por cualquier archivo macro no encontrado en el directorio actual. `<trayectoria>` puede ser cualquier nombre de trayectoria legal en el sistema operativo. Ejemplo: `asm56300 -Mfftlib trans.asm` Usa la convención de trayectorias de PC IBM y causa que el ensamblador busque en el subdirectorio `fftlib` del directorio actual por el archivo con el nombre del macro invocado encontrado en el archivo fuente, `trans.asm`.

Tabla A.5: Opciones del ensamblador (cont.)

Opción	Descripción
-V	Causa que el ensamblador reporte el progreso del ensamblado a la cadena de salida estándar.
-Z	causa que el ensamblador retire la información de símbolos del archivo de carga absoluto. Normalmente la información de símbolos es retenida en el archivo objeto con el propósito de permitir referencias simbólicas. Esta opción es válida sólo con las opciones **-A** y **-B** Nota: Se pueden usar opciones múltiples. Una cadena típica puede ser como sigue: Ejemplo: `asm56300 -A -B -L -G nombre de archivo.asm`

Control del ensamblado

Las directivas usadas para el control del ensamblado se muestran enseguida:

```
COMMENT     Inicia líneas de comentarios
DEFINE      Define cadena de substitución
END         Final del programa fuente
FAIL        Mensaje de Error generado por el
            programador
FORCE       Operador de puesta en modo forzado
HIMEN       Puesta de límites de memoria alta
INCLUDE     Incluir archivo secundario
LOMEM       Puesta de límites de memoria baja
MODE        Cambia modo  de relocalización
MSG         Mensaje generado por el programador
ORG         Inicializa espacio de memoria y
            localización de contadores
RADIX       Cambia la base (radical) de entrada
            para constantes
RDIRECT     Remueve directiva o mnemónico de tabla
SCSJMP      Puesta modo de control de brinco
            estructurado
SCSREG      Reasigna modo de control de brinco
            estructurado
UNDEF       Des-define el símbolo DEFINE
WARN        Aviso (alerta) generado por el
            programador
```

Definición de símbolos

Las directivas usadas para la definición de símbolos de control son:

```
ENDSEC      Fin de sección
EQU         Hacer igual símbolo a valor
GLOBAL      Declaración de sección de símbolos globales
GSET        Puesta de símbolo global a valor
LOCAL       Declaración de sección de símbolos locales
SECTION     Inicia sección
SET         Puesta de símbolo a valor
XDEF        Sección de definiciones de símbolos externos
XREF        Sección de referencias de símbolos externos
```

Definición de datos/Asignación de almacenamiento

Las directivas para controlar la definición de datos constantes y asignar almacenamiento se muestra a continuación. Se ha escrito entre paréntesis la definición en inglés de algunos términos, porque resulta más fácil de recordar la directiva por su definición en inglés, por ejemplo DS = Define storage.

```
BADDR       Puesta del espacio (buffer) de direcciones
BSB         Bloque de almacenamiento con inversión de bit
            (Block storage bit-reverse)
BSC         Bloque de almacenamiento de constantes
            (Block storage of constant)
BSM         Bloque de almacenamiento tipo módulo
            (Block storage modulo)
BUFFER      Inicia espacio de almacenamiento
DC          Define constante
            (Define constant)
DCB         Define constante tipo byte
            (Define constant byte)
DS          Define almacenamiento
            (Define storage)
DSM         Define almacenamiento tipo módulo
            (Define modulo storage)
DSR         Define almacenamiento tipo acarreo inverso
            (Define reverse carry storage)
ENDBUF      Fin de espacio de almacenamiento
```

Control de listado y opciones

Las directivas de control del listado de salida se muestran a continuación:

```
LIST        Listar el ensamblado
LSTCOL      Ajuste del ancho del campo de listado
NOLIST      Detiene el listado del ensamblado
OPT         Opciones del ensamblador
PAGE        Tamaño/página superior del paginado
PRCTL       Envía la cadena de control a la impresora
STITLE      Inicializa subtítulo del programa
TABS        Ajusta fin de tabulador en el listado
TITLE       Inicializa título de programa
```

Control del archivo objeto

Las directivas de control del archivo objeto son:

```
COBJ        Comenta código objeto
```

```
IDENT       Registro de identificación del código objeto
SYMOBJ      Escribe información de símbolos al archivo
            objeto
```

Macros y ensamblado condicional

Las directivas para macros y ensamblado condicional son:

```
DUP         Duplicar secuencia de líneas fuente
DUPA        Duplicar secuencia con argumentos
DUPC        Duplicar secuencia con caracteres
DUPF        Duplicar secuencia en lazo
ENDIF       Fin de ensamblado condicional
EDNM        Fin de definición de macro
EXITM       Salir de macro
IF          Directiva de ensamblado condicional
MACLIB      Biblioteca macro
MACRO       Definición de macro
PMA MACRO   Purgar definición de macro
```

Programación estructurada

Las directivas para programación estructurada son:

```
.BREAKD     Salir de construcción de lazo estructurado
.CONTINUE   Continuar siguiente iteración de lazo
            estructurado
.ELSE       Ejecuta los enunciados siguientes cuando
            .IF es falso
.ENDF       Final de lazo .FOR
.ENDI       Final de condición .IF
.ENDL       Final de lazo por alambrado o por
            circuitería (hardware)
.ENDW       Final de lazo .WHILE
.FOR        Comienza lazo .FOR
.IF         Comienza condición .IF
.LOOP       Comienza lazo alambrado (o por hardware)
.REPEAT     Comienza lazo de repetición
.UNTIL      Final de lazo .REPEAT
.WHILE      Comienza lazo .WHILE
```

Algunas de estas directivas son conocidas en otros lenguajes, por lo tanto no se da una explicación exhaustiva de su uso. Por ejemplo, un lazo WHILE es aquel que se ejecuta mientras se cumple cierta condición.

A.3.4. Ensamblado del programa de ejemplo

El ensamblador es un programa basado en el sistema operativo MS-DOS; por lo tanto, para usar el ensamblador debe salir del ambiente Windows, o abrir una ventana con el icono MS-DOS. Para ensamblar el programa de ejemplo introduzca: '+asm56300 -a -b -l -g ejemplo.asm+ en el directorio evm30xw creado durante el proceso de instalación dado en la sección A.1.2. Esto crea dos archivos adicionales: `ejemplo.cld` y `ejemplo.lst`. El archivo `ejemplo.cld` es un archivo objeto absoluto del programa, él es cargado en el DSP56303. El archivo ejemplo.lst es el archivo de listado; da todos detalles completos de en dónde está ubicados el programa y los datos en la memoria del DSP56303.

A.4. Encadenador DSP de Motorola

Aun cuando no es necesario para el ejemplo sencillo, el encadenador DSP también está incluido con el DSP56303EVM. El encadenador DSP es un programa que procesa archivos objetos relocalizables producidos por el ensamblador DSP y genera un archivo absoluto ejecutable el cual puede ser cargado al DSP56303. El encadenador DSP se incluye en el CD *Motorola Tools* y puede ser instalado siguiendo las instrucciones de la sección A.1.2. El formato general de la línea de mandos para invocar el encadenador es

`dsplnk[opciones]<nombres-de-archivos>`

donde `dsplnk` es el nombre del programa encadenador DSP,
y `<nombres-de-archivos>` es una lista de archivos objeto relocalizables a ser encadenados.

A.5. Opciones del encadenador

Las opciones del encadenador no se incluyen en este texto y tampoco se incluyen las directivas del encadenador. La información sobre las opciones del encadenador se pueden encontrar en el *DSP56303EVM User's Manual*, documento: DSP56303EVMUM/D.

A.6. Introducción al programa depurador

En esta sección se describe brevemente el depurador (*debugger*) de Domain Technologies, dando solamente los detalles requeridos para trabajar este ejemplo. Para los detalles completos sobre el depurador y un tutorial informativo, consulte el *Debug-56K Manual*. El depurador de Domain Technologies viene incluido con el DSP56303EVM en el CD-ROM de Domain Technologies, y puede ser instalado siguiendo las instrucciones en línea. Si

está usando Windows95 o Windows NT, el programa de instalación será lanzado automáticamente cuando inserte el CD en el lector. Para usar el depurador, haga un doble *click* sobre el icono etiquetado *evm30xw* en el grupo de programas creado cuando se instaló el depurador.

La forma de desplegado del depurador es similar a la mostrada en la figura A.4; la pantalla está dividida en cuatro ventanas: la ventana de mandos (*command*), la ventana de datos, la ventana de desensamblado (*unassembly*), y la ventana de registros. La ventana de mandos es la ventana activa (o seleccionada), lo cual significa que las ordenes de las teclas serán puestas en el ventana de mandos. La ventana de datos despliega los datos del DSP56303. La ventana de desensamblado muestra los programas de DSP56303 resaltando la siguiente instrucción a ser ejecutado. La ventana de registros muestra el contenido de los registros internos del DSP56303.

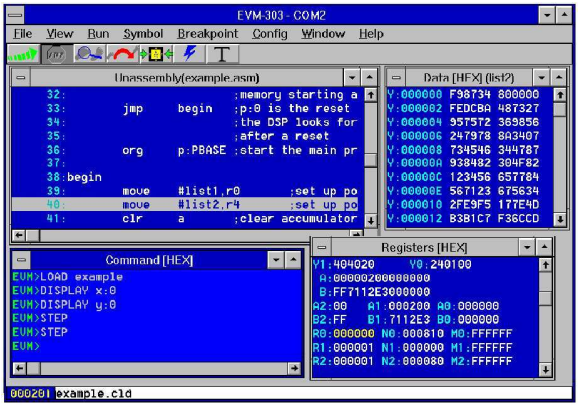

Figura A.4: Ejemplo de la ventana del depurador.

Con la ventana de mandos seleccionada como en la figura A.4, la barra de herramientas en la parte superior de la pantalla cambiará y mostrará los botones para los mandos más empleados en la ventana de mandos. De izquierda a derecha los mandos son `go`, `stop`, `step`, `jump`, `automatic update`, `reset`, y `radix` (una traducción aproximada para cada término sería la siguiente: arrancar, detener, paso, ir a, actualización automática, reiniciar, y base).

- `Go` corre el programa en el DSP56303 a partir de la dirección del contador de programa.
- `Stop` detiene al DSP56303.
- `Step` ejecuta una sola instrucción.
- `Jump` es similar a step, excepto que las subrutinas son tratadas como una instrucción.

- `Automatic update` enciende el modo de actualización en pantalla automática, de modo que el DSP56303 es interrumpido periódicamente para actualizar las ventanas de datos y de registros.
- `Reset` reinicia al DSP56303.
- `Radix` puede usarse para cambiar la base (binario, hexadecimal, etc.) de la ventana seleccionada.

Cuando se seleccionan otras ventanas aparecen otros botones, y funcionan como se describe en el *Manual del Debug-56k*, el cual está contenido en el CD-ROM de Domain Technologies.

A.6.1. Ejecución del programa

Para cargar el programa de ejemplo en el depurador, seleccione con el ratón la ventana de mandos y teclee `load ejemplo`. La instrucción en la línea 33 será resaltada en la ventana *unassembly* debido a que es la primera instrucción a ser ejecutada. Pero, antes de ejecutar el programa, verifique que los valores esperados en la ventana de datos estén ahí. Para hacer esto teclee `display x:0` y `display y:0`. Los datos son mostrados en la ventana de datos.

Para ejecutar por pasos teclee `step` en la ventana de mandos. Para seguir un atajo haga *click* sobre el botón `step` o teclee el inicio del mando y presione la barra espaciadora y el depurador completará el resto del mando. Para repetir el último mando presione `return` o *intro*. En tanto se vaya avanzando en cada paso a través del código, note que los registros en la ventana de registros son cambiados por las instrucciones. Después de cada ciclo cualquier registro que haya sido cambiado será resaltado. Una vez que haya ejecutado por pasos el programa, asegúrese que el programa se haya ejecutado correctamente comprobando que el resultado en el acumulador `a` sea $FE9F2051 6DFCC2.

Ejecutar por pasos, como en este caso, es bueno para programas cortos, pero es impráctico para programas complejos y largos. La forma de depurar programas largos es usar puntos de ruptura, que son puntos definidos por el usuario donde la ejecución del código se detiene, permitiendo al usuario ejecutar por pasos a través de la sección de interés. En el ejemplo ponga un punto de ruptura para verificar que los valores en `r0` y `r4` son correctos antes del lazo `do`, teclee `break p:\$106` en la ventana de mandos. La línea antes del lazo será resaltada en la ventana *unassembly*, indicando que se ha puesto un punto de ruptura. Para apuntar el DSP56303 al inicio del programa teclee `change pc 0`. Esto cambia el contador de programa de modo que apunte al vector de reinicio. Para correr el programa teclee `go`. El DSP56303 se detiene cuando alcanza el punto de ruptura, y se puede ir por pasos a través del resto del código.

Para salir del depurador, teclee `quit` en la ventana de mandos.

A.7. Resumen técnico del DSP56303EVM

A.7.1. Descripción y características del DSP56303EVM

En el folleto del producto (DSP56303EVMP/D), incluido en paquete (*kit*), se da una descripción del DSP56303EVM. Las principales características son las siguientes:

- Procesador digital de señales de 24-bit DSP56303.
- FSRAM para expansión de memoria y memoria Flash PEROM para operación independiente.
- Codec de audio con calidad de CD de 16 bits.
- Circuitería para convertidor de mandos.

A.7.2. Descripción del DSP56303

Una descripción completa del DSP56303, incluyendo información funcional y para el usuario, se da en los siguientes documentos:

- **DSP56303 Technical Data** (**Número de orden del documento DSP56303/D**): da una lista de caraterísticas y especificaciones incluyendo descripción de señales, requisitos de potencia, requerimientos de tiempos de AC, y empaquetados disponibles.
- **DSP56303 User's Manual (Número de orden de documento DSP56303UM/AD**: da descripción del DSP e información detallada acerca de los componentes en el CI incluyendo los mapas de memoria I/O, funcionalidad de periféricos, y registros de control y de estado (*status*) para cada subsistema.
- **DSP56300 Family Manual (Número de orden del documento DSP56300FM/AD**): da una descripción detallada del núcleo del procesador incluyendo estado interno y registros de control y descripción del juego de la familia de instrucciones.

Refiérase a estos documentos para información detallada acerca de la funcionalidad del C.I. y operación. Estos documentos estarán incluidos en el *kit* ya sea en CD o en forma impresa.

Nota: También se incluye una lista detallada de errores conocidos del C.I. Vea el documento *DSP56303 Chip Errata* para información que haya cambiado desde la publicación del documento de referencia listado previamente. Se puede obtener la última versión en el sitio web:
`http://www.mot.com/SPS/DSP/chiperrata/index.html`

A.7.3. Memoria

El DSP56303EVM incluye la siguiente memoria externa:

- 64K×24-bit de RAM (FSRAM) para expansión de memoria.
- 128K×8-bit de memoria flash para operación independiente.

En la figura A.5 se muestra la localización de la memoria FSRAM y Flash sobre el DSP56303EVM. La figura A.6 muestra un diagrama de bloques funcionales del DSP56303EVM incluyendo los dispositivos de memoria.

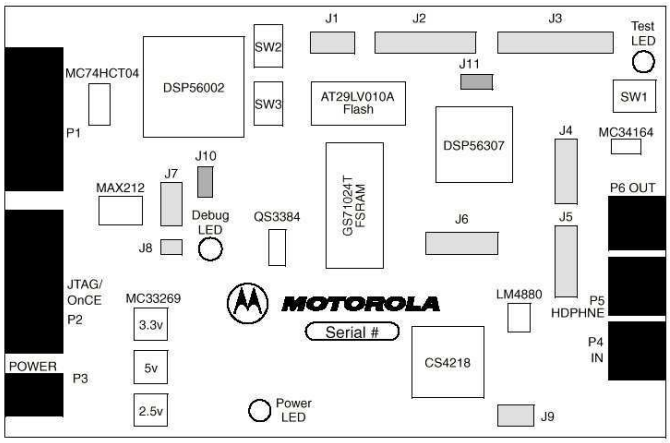

Figura A.5: Ubicación de componentes del DSP56303EVM.

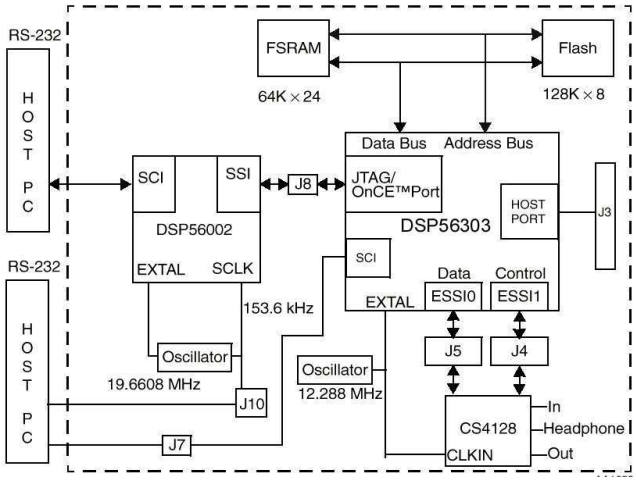

Figura A.6: Diagrama funcional a bloques del DSP56303EVM.

FSRAM

El DSP56303EVM usa un banco de RAM estática rápida de 64K×24-bit (GS71024T-10, etiquetada como U4) para expansión de memoria. La GS71024T-10 usa una sola fuente de poder de 3.3 V y tiene un tiempo de acceso de 10 ns. Las siguientes secciones detallan la operación de FSRAM.

Conexiones de la FSRAM

La conexión básica para la SFRAM se muestra en la figura A.7. Las terminales de entrada/salida de datos IO0-IO23 para la SFRAM se conectan a las terminales D0-D23 del DSP56303. Las líneas de habilitado de escritura ($\overline{WE}$) y de habilitado de salida ($\overline{OE}$) se conectan a las líneas de escritura ($\overline{WE}$)y de lectura ($\overline{RD}$) del DSP56303, respectivamente. La señal de habilitado de la FSRAM ($\overline{CE1}$) es generada por la línea de atributo de dirección 0 (AA0) del DSP56303. La actividad del FSRAM es controlada por AA0 y el registro de atributo de dirección 0 (AAR0) correspondiente. Las terminales de entrada de direcciones del FSRAM, A0-A15, están conectadas respectivamente a las terminales del puerto de direcciones A del DSP. Esta configuración selecciona un mapa unificado de memoria de 64K palabras. La memoria unificada no contiene memoria particionada de datos X, datos Y, y programa; por lo tanto, el acceso a P:$1000, X:$1000, y Y:$1000 es tratado como acceso a la misma celda de memoria y no son posibles movimientos (o transferencias) de 48-bit de memoria de datos a, o desde, la memoria externa FSRAM.

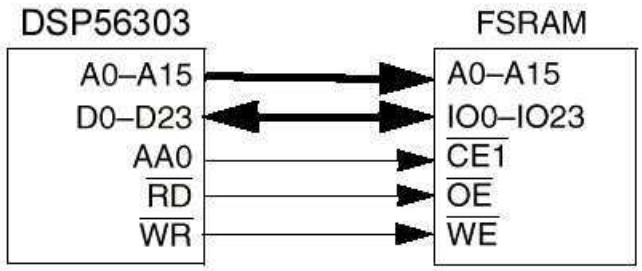

Figura A.7: Conexiones de la FSRAM al DSP56303.

Ejemplo: Programación de AAR0

Como se mencionó anteriormente, la actividad de la FSRAM está controlada por la terminal AA0 del DSP56303 y el correspondiente AAR0. AAR0 controla el tipo de acceso externo, el tipo de memoria y cuál dirección externa de memoria tendrá acceso la FSRAM. La figura A.8 muestra el mapa de memoria que está ligada a los ajustes de AAR0 descritos en este ejemplo.

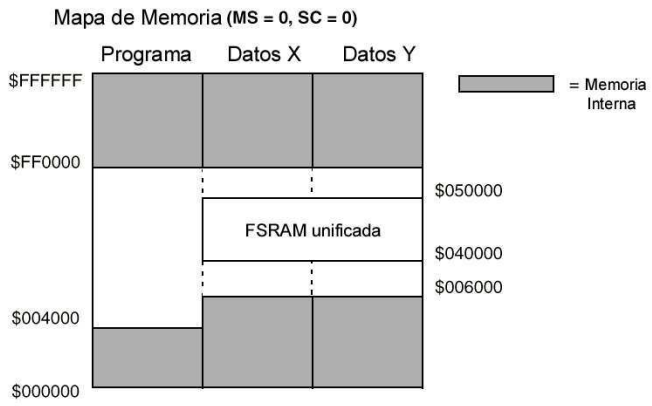

Figura A.8: Ejemplo de mapa de memoria con la memoria externa unificada.

Nota: En este ejemplo, el bit de conmutación de memoria en el registro de modo de operación (OMR) es borrado y el bit de compatibilidad de 16-bit en el registro de estado es borrado.

En la figura A.8 la FSRAM responde a los 64K de las direcciones de memoria de datos X y Y entre $040000 y $04FFFF. Sin embargo, con el mapa de memoria unificado, los accesos a la misma localidad de memoria externa son tratados como accesos a la misma celda de memoria.

Existe un mecanismo de prioridad entre los cuatro registros de control AAR. AAR3 tiene la más alta prioridad y AAR0 tiene la más baja. El bit 14 del OMR, el bit de deshabilitado de prioridad de direcciones (APD) controla cuáles terminales AA son declaradas válidas cuando ocurre un conflicto de selección (es decir, la dirección externa iguala la dirección y el espacio que está especificado en más de un AAR). Si se borra el bit APD cuando ocurre un conflicto de selección, sólo los terminales de más alta prioridad de AA son declaradas válidas. Si el bit APD está puesto cuando ocurre un conflicto de selección, las terminales de más baja prioridad de AA son declarados además de los terminales de más alta prioridad de AA. Para este ejemplo, sólo una terminal de AA debe ser declarada, AA0. Entonces, el bit APD puede borrarse.

La figura A.9 muestra los ajustes de AAR0 para este ejemplo. Los bits del tipo de acceso externo (BAT1 y BAT0) son puestos a 0 y 1, respectivamente, para denotar el acceso FSRAM. El bit de polaridad de atributo (BAAP) es borrado para definir AA0 como bajo activa (activa en nivel bajo). El multicanalizador de direcciones no es necesario con la FSRAM; por lo tanto, se borra el bit de multicanalización de direcciones BAM. El empaquetado no es necesario con la FSRAM; así el bit de habilitación de empaquetamiento BPAC es borrado para deshabilitar esta opción.

Los bits de habilitado de los espacios P, X, y Y (BPEN, BXEN, BYEN)

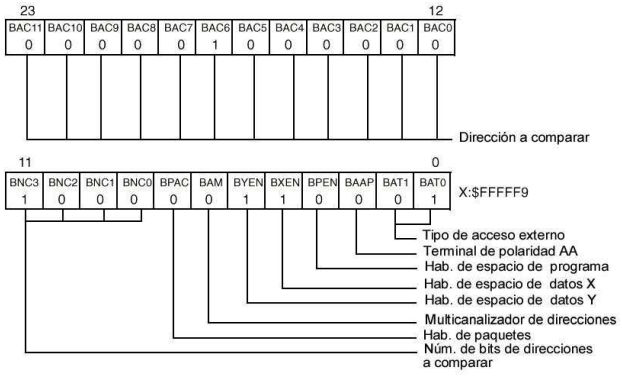
23
12
BAC11 BAC10 BAC9 BAC8 BAC7 BAC6 BAC5 BAC4 BAC3 BAC2 BAC1 BAC0
0 0 0 0 0 1 0 0 0 0 0 0
Dirección a comparar
11
0
BNC3 BNC2 BNC1 BNC0 BPAC BAM BYEN BXEN BPEN BAAP BAT1 BAT0
1 0 0 0 0 0 1 1 0 0 0 1
X:$FFFFF9
Tipo de acceso externo
Terminal de polaridad AA
Hab. de espacio de programa
Hab. de espacio de datos X
Hab. de espacio de datos Y
Multicanalizador de direcciones
Hab. de paquetes
Núm. de bits de direcciones
a comparar

Figura A.9: Registro de atributos de direcciones AAR0.

definen si la FSRAM es habilitada durante el acceso externo a los espacios P, de datos X, o datos Y, respectivamente. Para este ejemplo, los bits BXEN y BYEN están puestos, y BPEN está borrado para permitir a la FSRAM responder sólo a los accesos de memoria de datos X y Y.

El número de bits de direcciones a comparar BNC(3:0) y los bit de dirección a comprar (BAC(11:0) determinan cuál dirección de memoria externa tienen acceso a FSRAM. Los bits BNC definen el número de bits de dirección superiores que son comparados entre los bits BAC y la dirección externa para determinar si se tiene acceso la FSRAM. Para este ejemplo, la FSRAM es asignada para responder a la direcciones entre \$040000 y \$04FFFF. Entonces, los bits BNC son puestos a \$8 y los bits BAC son puestos a \$040. Si los 8 bits más significativos de la dirección externa son 00000100, se tiene acceso a la FSRAM.

A.7.4. Flash

El DSP56303EVM usa un C.I. Atmel AT29LV010A-20T (U3) para tener 128K×8-bit Flash CMOS para operación independiente (es decir, operación de arranque sin accesar el DSP56303 a través del puerto JTAG/OnCE). La AT29LV010 usa un fuente de potencia de 3.3 V y tiene un tiempo de acceso de 200 ns.

Conexiones de la Flash

La conexión básica de la Flash se muestra en la figura A.10.

Las terminales de direcciones (A0-A16) de la flash se conectan a las terminales de direcciones respectivas del puerto A en el DSP. Las terminales de entrada/salida de datos de la flash I/O0-IO7 son conectadas a las terminales D0-D7 del DSP56303. Las líneas de habilitado de escritura ($\overline{WE}$) y de habilitado de salida ($\overline{OE}$) se conectan a las líneas de habilitado de escritura

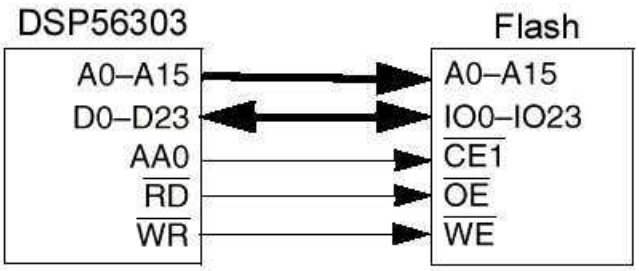

Figura A.10: Conexiones de la Flash.

($\overline{WE}$) y lectura ($\overline{RD}$) respectivamente. La línea de atributo de direcciones 1 (AA1) genera la señal de habilitado del C.I (es decir del flash) ($\overline{CE}$). Esta línea es mejor conocida por su denominación en inglés, "chip enable".

A.7.5. Programación para operación independiente

Las terminales de modo del DSP56303 determinan el modo de operación y procedimiento de arranque cuando el DSP56303 sale del estado de reinicio. El interruptor en SW1 reinicia al DSP56303 declarando y luego borrando la terminal $\overline{RESET}$ del DSP56303. Las terminales de modo MODA, MODB,MODC, y MODD son muestreadas mientras el DSP56303 sale del estado de reinicio. Las terminales de modo del DSP56303EVM están controladas por el bloque de puentes J1 mostrado en la figura A.5. El DSP56303 arranca desde la Flash después del reinicio si hay puentes conectando las terminales 3-4 y 5-6 sobre J1 (Modo 1: MODA y MODD están puestas, y MODB y MODC están borradas).

A.8. Codec de Audio

La sección analógica del DSP56303EVM usa un CS4218-KQ de Crystal Semiconductor para dos canales de conversión A/D de 16 bits y dos canales de conversión D/A de 16 bits. Observe la figura A.5 para la localización del codec (CS4218) sobre el DSP56303EVM y la figura A.6 para el diagrama funcional de bloques del codec dentro del módulo de evaluación. El CS4218 usa una fuente de poder digital de 3.3 V y una fuente de poder analógica de 5 V.

El CS4218 es controlado por una señal de 12.288 MHz en la terminal de entrada del reloj maestro (CLKIN). El oscilador en Y1 crea una señal de 5 V a 12.288 MHz. El QS3384 en U5 entonces convierte la señal de 5 V a 3.3 V como entrada a la terminal CLKIN del codec y la entrada EXTAL del DSP56303. Para más información vea a la hoja de datos del CS4218 incluida en el *kit*.

El CS4218 es muy flexible, ofrece frecuencias de muestreo que se pueden seleccionar entre 8 kHz y 48 kHz. La frecuencia de muestreo se selecciona usando puentes sobre el bloque de puentes J9. La tabla A.6 muestra la posición de los puentes que seleccionan las posibles frecuencias de muestreo para el DSP56303EVM.

Tabla A.6: Selección de frecuencias de muestreo del CS4218

Term. 1-2 J9 (MF6)	Term. 3-4 J9 (MF7)	Term. 5-6 J9 (MF8)	Frecuencia de muestreo (kHz)
Puente	Puente	Puente	48.0
Puente	Puente	Abierto	32.0
Puente	Abierto	Puente	24.0
Puente	Abierto	Abierto	19.2
Abierto	Puente	Puente	16.0
Abierto	Puente	Abierto	12.0
Abierto	Abierto	Puente	9.6
Abierto	Abierto	Abierto	8

El codec está conectado al ESSI0 del DSP56303 a través de los puentes J4 y J5 mostrados en la figura A.5. El bloque de puentes J4 conecta las terminales del ESSI1 del DSP56303 a las terminales de control del CS4218. El bloque de puentes J5 conecta las terminales del ESSI0 del DSp56303 a las terminales de datos del CS4218. Retirando estos puentes, el usuario tiene acceso completo a las terminales ESSI0 y ESSI1 del DSP56303. La siguiente sección describe las conexiones para las secciones analógica y digital del codec.

A.8.1. I/O analógicas del codec

El DSP56303EVM contiene conectores hembra de 1/8 de pulgada para entrada estereo, salida estereo, y audífonos. La figura A.11 muestra la circuitería analógica del codec.

El conector estereo marcado como P4/IN en el DSP56303EVM se conecta a la terminales de entrada derecha e izquierda RIN2 y LIN2 del codec. Los niveles estándar de entrada son de 2 Vpp y el codec requiere que los niveles de entrada sean limitados a 1 Vpp. Por lo tanto, un divisor de tensión forma un atenuador de 6 dB entre P4 y el CS4218.

Las terminales de salida del canal izquierdo y derecho del codec, ROUT y LOUT, proveen sus señales analógicas de salida, a través del conector estereo marcado como P6/OUT sobre el DSp56303EVM. Las salidas del codec también están conectadas al conector estereo marcado como P5/HDPHNE sobre el DSP56303EVM a través del amplificador de audio de potencia de

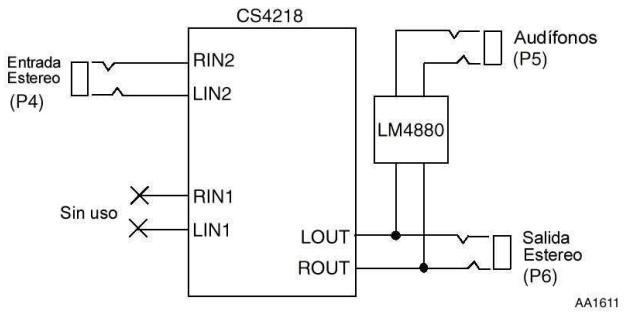

Figura A.11: Diagrama de entrada/salida analógicas del codec.

National Semiconductor's LM4880 en U8. El conector estereo para audífonos permite la conexión directa de audífonos al DSP56303EVM.

A.8.2. Interfase digital del codec

La figura A.12 muestra la interfase digital del codec. La tabla A.7 y la tabla A.8 muestran las selecciones de los puentes para Habilitar/Deshabilitar las señales del codec.

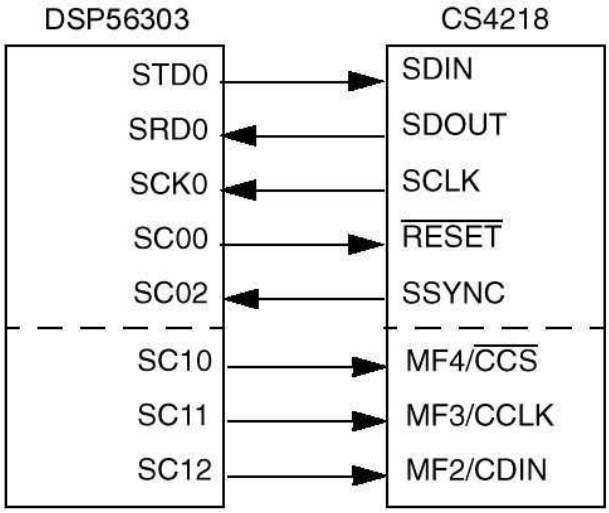

Figura A.12: Conexiones de la interfase digital del codec.

La interfase serie del codec transfiere datos de audio digital y datos de control hacia y desde el dispositivo. El codec se comunica con el DSP56303 a través del ESSI0 para información de datos y a través del ESSI1 para información de control. El codec tiene tres modos de operación serie que se

Tabla A.7: Opciones del bloque de puentes JP5

JP5	Señal del DSP	Señal del codec
1-2	SCK0	SCLK
3-4	SC00	RESET
5-6	STD0	SDIN
7-8	SRD0	SDOUT
9-10	SC01	–
11-12	SC02	SSYNC

Tabla A.8: Opciones del bloque de puentes JP4

JP4	Señal del DSP	Señal del codec
1-2	SCK1	–
3-4	SC10	CCS
5-6	STD1	–
7-8	SRD1	–
9-10	SC12	CDIN
11-12	SC11	CCLK

pueden seleccionar por las terminales de selección de modo serie SMODE1, SMODE2, y SMODE3. Las terminales SMODE del DSP56303 se seleccionan para habilitar el modo serie 4, el cual separa los datos de audio de los datos de control. Las terminales SMODE también se ajustan para habilitar el submodo maestro con bloques (frames) de 32-bits, siendo los primeros 16 bits el canal izquierdo, y los segundos 16 bits son el canal derecho.

El ESSIO del DSP56303 transfiere los datos de información a y desde el codec. La terminal de transmisión de datos serie (STD0) del DSP56303 transmite datos al codec. La terminal de recepción de datos serie (SRD0) del DSP56303 recibe datos desde el codec. Estas dos terminales están conectadas a las terminales del puerto de entrada de datos serie del codec (SDIN) y del puerto de salida de datos serie (SDOUT), respectivamente. En el submodo maestro, la terminal del reloj del puerto serie del codec (SCLK) da la cadencia (*bit rate*) del reloj serie para la interfase ESSI0. Éste está conectado a la terminal bidireccional del reloj serie (SCK0) del DSP56303. La terminal de control serie 0 (SC00) del DSP56303 está programado para controlar la señal de reinicio del codec $\overline{RESET}$. La terminal de control serie 2 (SC02) está conectada a la señal de sincronía del puerto serie del codec (SSYNC). Un flanco de subida sobre SSYNC indica que está por comenzar un nuevo bloque (frame).

Las terminales ESSI1 del DSP56303 son usadas como señales I/O de

propósito general (GPIO) para transferir los datos de control al codec. Los datos de control necesitan ser transferidos sólo cuando cambian. La terminal de control serie 0 (SC10) del DSP56303 está programada para controlar la terminal multi-función 4 del codec o la terminal de selección de control de datos del C.I., $MF4/\overline{CCS}$. Esta terminal debe estar en estado bajo para entrar en modo control de datos. La terminal de control serie 1 (SC11) se conecta a la terminal multi-función 3 del codec o la terminal de reloj de control de datos (MF3/CCLK). Los datos de control son introducidos sobre el flanco de subida de CCLK. La terminal de control serie 2 (SC12) está conectada a la terminal multi-función 2 o la terminal de entrada de datos de control, MF2/CDIN. Esta terminal contiene los datos de control para el codec.

A.9. Convertidor de mandos

El DSP56303 usa el DSP56002 para realizar la conversión de mandos JTAG/OnCE. La interfase de comunicación serie (SCI) del DSP56002 se comunica con la PC anfitrión a través de un conector serie RS-232. El SCI del DSP56002 recibe los mandos desde la PC. El conjunto de datos puede incluir leer datos, escribir datos, reiniciar el módulo OnCE, reiniciar el DSP56303 (entonces se usa la terminal HA2 del DSP56002 para reiniciar el DSP56303), petición del módulo OnCE o liberar al módulo OnCE. El programa de conversión de mandos del DSP56002 interpreta loa mandos recibidos por desde la PC y envía una secuencia de instrucciones al puerto JTAG/OnCE del DSP56303. El DSP56303 entonces puede continuar recibiendo datos o puede transmitir datos hacia el DSP56002. El DSP56002 envía una respuesta (*replay*) a la PC para dar información del estado. El conjunto de respuestas puede incluir confirmación afirmativa (*acknowledge good*), confirmación negativa (*acknowledge bad*), en modo depurador (*in debug mode*), fuera de modo depurador (*out of debug mode*), o lectura de datos (*data read*). Cuando el DSP56303 está en estado depurador, se ilumina el diodo emisor de luz rojo (LED2) de depurador.

El DSP56002 se conecta al puerto JTAG/OnCE del DSP56303 a través de un puente sobre J8. La tabla A.9 muestra las opciones de habilitado/deshabilitado del JTAG. El puente debe estar presente en J8 para usar el DSP56002 como convertidor de mandos. Observe a la figura A.5 para la localización de J8 sobre el DSP56303EVM y a la figura A.6 para ver el diagrama funcional. La figura A.13 muestra el diagrama de la interfase serie RS-232. La tabla A.10 muestra la terminales de conector RS-232, (P2).

Para transmitir los datos entre la PC y el DSP56002 se usa un transceptor RS-232 de Maxim MAX212 alimentado por 3 V, en U11. Los datos serie son transmitidos desde la señal TXD (*transmitted data*) de la PC y recibidos en el DSP56002 por la terminal RXD (*receive data*). Similarmente, son trasmitidos datos serie desde la señal TXD (*transmit data*) del DSP56002

Tabla A.9: Opciones de Hab/DesHab de JTAG

J8	Opción seleccionada
1-2	Convertidor de mandos de la tarjeta habilitado
abierto	Convertidor de mandos de la tarjeta des-habilitado

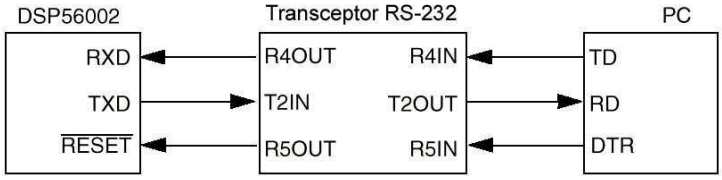

Figura A.13: Interfase serie RS232.

y recibidos por la señal RD (*received data*) de la PC. La terminal *Data Terminal Ready* (DTR) declara la terminal $\overline{RESET}$ del DSP56002.

Como una opción, la terminal 14 del conector JTAG/OnCE del DSP56303 EVM en J6 permite al usuario conectar una tarjeta convertidora de mandos ADS, directamente al DSP56303EVM si no se usa el programa de conversión de mandos del DSP56002 (con el puente en J8 removido). La terminal 8 de J6 ha sido removida para que no se pueda conectar en forma incorrecta el cable al DSP56303EVM. La tabla A.11 muestra información de las terminales del conector JTAG/OnCE (J6). El cable JTAG desde el convertidor de mandos ADS está similarmente codificado de modo que no se pueda conectar incorrectamente al DSp56303EVM.

A.10. Interfase externas

El DSP56303EVM provee interfases externas con dispositivos no incluidos en la tarjeta por medio de sus puertos periféricos incluidos (*on-chip peripherals*). La mayoría de los puertos del DSP están conectados a conectores (de varios alfileres o puntas) sobre el módulo de evaluación (EVM) para facilitar el acceso directo a estas terminales por medio del uso de conectores o puentes.

A.10.1. El puerto de comunicación serie (SCI)

La conexión al puerto SCI del DSP puede hacerse en J7. En la tabla A.12 se indica la información de terminales. Las señales en J7 son señales de 3.3 V directas del DSP. Si se requieren señales con niveles RS-232 se deben instalar puentes en J7. Vea la tabla A.13 para enrutar las señales SCI del

Tabla A.10: Terminales del conector (P2) del depurador RS232

Número de terminal	Nombre de la señal de DSP	Número de terminal	Nombre de la señal del DSP
1	-	6	-
2	TxD	7	-
3	RxD	8	-
4	Reinicio	9	-
5	Tierra		

Tabla A.11: Terminales del conector (J6) del JTAG/OnCE

Número de terminal	Nombre de la señal de DSP	Número de terminal	Nombre de la señal del DSP
1	TDI	2	Tierra
3	TDO	4	Tierra
5	TCK	6	Tierra
7	-	8	Orientación.
9	PRESET	10	TMS
11	+3.3 V	12	-
13	DEZ	14	TRST

DSP a través de un convertidor de nivel RS-232 a P1. La información de salidas (terminales) de P1 se muestra en la tabla A.14.

Tabla A.12: Terminales del conector (J7) del SCI.

Número de terminal	Nombre de la señal de DSP	Número de terminal	Nombre de la señal del DSP
1	RxD	2	-
3	SCLK	4	-
5	TxD	6	-

Si se instala un puente en J10, el puerto SCI está controlado por el reloj de 153.6 kHz del oscilador de la tarjeta en lugar de un reloj externo via el conector del puerto serie (P1) o, internamente, por un temporizador del SCI. Si se instala J10 se deben remover el puente 3-4 de J7.

Tabla A.13: Opciones de los puentes en J7.

J7	Señal del DSP
1-2	RxD
3-4	SCLK
5-6	TxD

Tabla A.14: Terminales del conector (P1) del puerto serie del DSP.

Número de terminal	Nombre de la señal de DSP	Número de terminal	Nombre de la señal del DSP
1	-	6	-
2	TxD	7	SCLK
3	RxD	8	-
4	-	9	-
5	Tierra		

A.10.2. Puerto serie síncrono mejorado 0 (ESSI0)

La conexión al puerto ESSI0 del DSP se puede hacer en J5. Vea la tabla A.15 para la información de las terminales.

Tabla A.15: Terminales del conector (J5) del ESSI0

Número de terminal	Nombre de la señal de DSP	Número de terminal	Nombre de la señal del DSP
1	SCK0	2	-
3	SC00	4	-
5	STD0	6	-
7	SRD0	8	-
9	SC01	10	-
11	SC02	12	-

A.10.3. Puerto serie síncrono mejorado 1 (ESSI1)

La conexión al puerto ESSI1 del DSP se puede hacer en J4. Vea la tabla A.16 para la información de las terminales.

Tabla A.16: Terminales del conector (J4) del ESSI1

Número de terminal	Nombre de la señal de DSP	Número de terminal	Nombre de la señal del DSP
1	SCK1	2	-
3	SC10	4	-
5	STD1	6	-
7	SRD1	8	-
9	SC12	10	-
11	SC11	12	-

A.10.4. Puerto *Host* (HI08)

La conexión al puerto HI08 se puede hacer en J3. Vea la tabla A.17 para la información de las terminales.

Tabla A.17: Terminales del conector (J3) del HI08

Número de terminal	Nombre de la señal de DSP	Número de terminal	Nombre de la señal del DSP
1	H0	2	H1
3	H2	4	H3
5	H4	6	Tierra
7	H5	8	H6
9	H7	10	Reinicio
11	HA0	12	HA1
13	HA2	14	HCS
15	HREQ	16	HDS
17	+3.3 V	18	HACK
19	HRW	20	Tierra

A.10.5. Control del conducto (bus) de expansión

La conexión a las señales de control del conducto de expansión del DSP se puede hacer en J2. Vea la tabla A.18 para la información de las terminales.

A.10.6. Selector del modo

La selección del modo de arranque para el DSP56303 se puede hacer por las selecciones de puentes sobre J1.

Tabla A.18: Terminales del conector (J2) del conducto de expansión

Número de terminal	Nombre de la señal de DSP	Número de terminal	Nombre de la señal del DSP
1	+3.3 V	2	RD
3	WR	4	BG
5	BB	6	BR
7	TA	8	BCLK
9	BCLK	10	CAS
11	CLKOUT	12	AA1
13	AA0	14	AA1
15	AA3	16	Tierra